REPO, DOCUMENTATION PAGE

AFRL-SR-BL-TR-98

0200

Public reporting burden for this collection of information is estimated to average 1 hour per response, includin gathering
and maintaining the data needed, and completing and reviewing the collection of information. Send comm llection of
information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate way, Suite
1204, Arlington, VA 22202-4302, and to the Office of management and Budget, Paperwork Reduction Project (l

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 30 Sep 97	3. REPORT TYPE AND DATES COVERED Final (01 Jun 93 – 31 Mar 97)

4. TITLE AND SUBTITLE An Integrated System for NC Machining of Multi-Patch Surfaces	5. FUNDING NUMBERS F49620-93-1-0419
6. AUTHORS Professor Deba Dutta	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Michigan Ann Arbor, MI 48109-2125	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFOSR/NM 110 Duncan Avenue, Room B-115 Bolling Air Force Base, DC 20332-8080	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release	12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

An integrated system for numerically controlled machining of parts made of multi-patch surfaces has been developed. The tool paths are based on practical metrics such as length and curvature and on accuracy of the manufactured part.

19980225 069

DTIC QUALITY INSPECTED 4

14. SUBJECT TERMS multi-patch, tool path	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

JOURNAL OF COMPUTER AIDED DESIGN
"in print"

An Integrated System for NC Machining of Multi-Patch Surfaces

R. Sarma and D. Dutta

Department of Mechanical Engineering
The University of Michigan
Ann Arbor, MI 48109

Abstract:

We describe an integrated system for NC machining of parts made of multi-patch surfaces. The system enables the user to generate tool paths based on practical metrics such as length, curvature and number of tool paths. We also describe two methods for interfacing with commercial CAD/CAM packages.

Keywords: *CAD, CAM, Parametric Surfaces, Tool Path Generation*

1.0 Introduction:

Milling is a commonly used machining method for manufacturing complex shapes such as the ones found in the automobile, aerospace, ship building and health care industries. While rapid prototyping is preferred for manufacturing soft tooling, where the production batch sizes are small, numerical control (NC) milling is the preferred method to manufacture hard tooling which can be reused and is more durable compared to the soft tooling.

Milling is the process of material removal due to the interaction of a cutting tool moving relative to the workpiece. The manufactured part is obtained as the tool comes in contact with the nominal designed part (henceforth referred to as part) along specific trajectories called tool paths. The accuracy and cost (proportional to the time of manufacture) of the manufactured part are affected by the geometry and spacing of tool paths on the part. Especially, the tool paths for finish machining are very important, since finishing cuts account for 50% of the total machining time and directly affect the accuracy or surface finish of the manufactured part.

In this paper, we introduce an integrated system for NC machining of parts made of multipatch surfaces. The system enables the user to generate tool paths based on practical metrics such as length and curvature of the tool paths and accuracy of the manufactured part. The details of the tool path generation scheme, for a single parametric patch, has been described in [Sarma and Dutta, 1996a]. In this paper we focus on extending the method to handle multiple patches, acommon occurence in CAD models of industrial parts. For this paper we assume that the individual patches comprising the part are four sided.

This paper is structured as follows. The rest of this section will be devoted to discussing the various techniques for tool path generation being used in research and commercial systems. In Section 2, we give an overview of our method for tool path generation for a single patch. The focus of this paper is on generating tool paths for parts made of multi-patch surfaces. This is described in Sections 3-5. In addition, we will discuss issues involved in the integration of our method with commercial CAD/CAM systems in Section 6. We conclude with Section 7.

1.1 Existing Tool Path Generation Techniques:

Tool path generation is the method by which the trajectory of the tool is constrained to lie on the part. This has been traditionally enforced by (a) constraining the tool path with the help of curves/surfaces defined in the object space of the part and (b) constraining the tool path by specifying curves in the parametric space of the part. Note that the literature quoted in the following paragraphs is by no means exhaustive due to the number of papers in the field of tool path generation.

Surface section curves [Bobrow, 1985] [Huang and Oliver, 1994] and projection curves [Unigraphics, 1992], often used in tool path generation fall in the former category. In the latter category are iso-parametric [Loney and Ozsoy, 1987] and iso-curvature curves [Jensen and Anderson, 1992]. Recently, offset curves on the part [Suresh and Yang, 1994] [Sarma and Dutta, 1996a] have been proposed as a method to generate tool paths.

A main disadvantage of object space methods is that the user only has control over the maximum scallop height, which is attained at discrete points on the manufactured part [Huang and Oliver, 1994] [Sarma and Dutta, 1996b]. Parameter space methods are advantageous since the user can have a direct control over the scallop height of the manufactured part [Suresh and Yang, 1994] [Sarma and Dutta, 1996a].

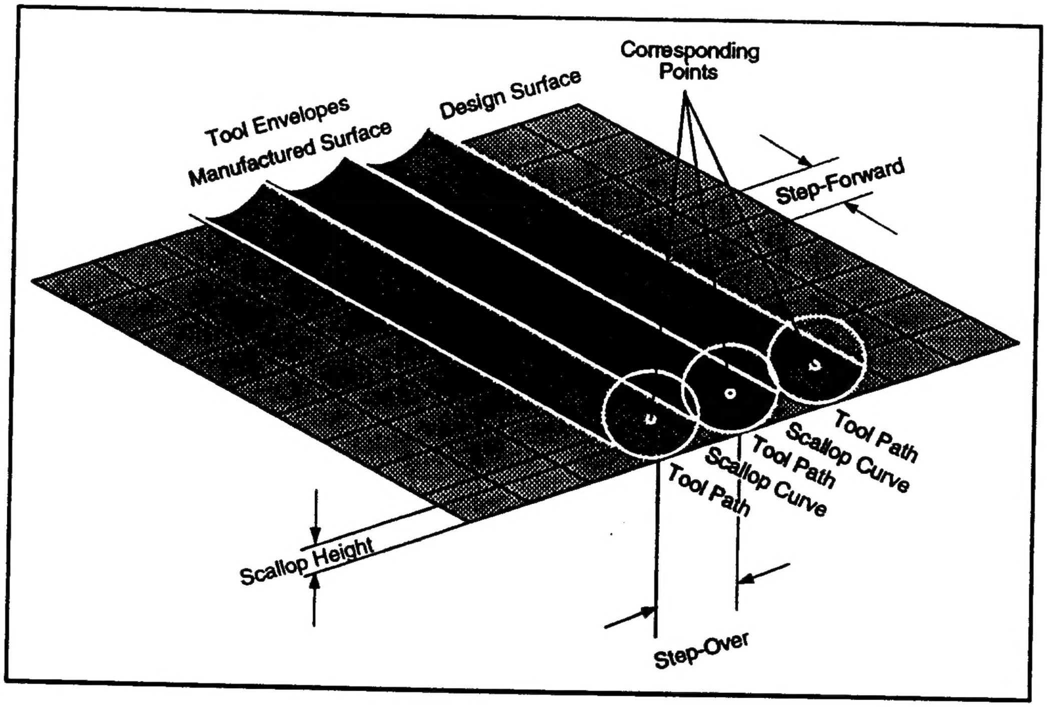

FIGURE 1. Illustration of Scallop Height, Step-Over etc.

The methods for tool path generation for compound surfaces, usually involve discretizing the part into a mesh of simpler elements [Lai and Wang, 1994] e.g. triangular elements etc. Of the methods mentioned above, [UniGraphics, 1992] and [Sarma and Dutta, 1996b] specifically discuss issues related to multi-patch machining. The extension of tool path generation techniques to multi-patch surfaces is non-trivial and necessary for usage in real world problems.

1.2 Commercial Systems for Tool Path Generation:

Several commercial systems [UniGraphics, 1992] [Catia, 1994] [ProEngineer, 1995] provide the capability for tool path generation. Tool paths used include plane section curves, iso-parametric curves, projection curves and offset curves. The method for calculating the step-over (illustrated in Figure 1) involves specifying (a) the maximum step-over distance in object space, (b) the maximum scallop height and (c) the number of tool paths.

Our method for tool path generation [Sarma and Dutta, 1996a] provides several new options for the user to generate tool paths based on practical metrics such as length, curvature and number of tool paths. Such metrics, which give the user valuable information related to productivity (time and accuracy of machining), are not currently available in commercial CAD/CAM systems. The integration of our method for tool path generation with commercial CAD/CAM systems is expected to supplement the existing methods for tool path generation and provide more alternatives to the user for geometrically complicated parts.

2.0 Overview of Tool Path Generation for a Single Patch:

For completeness, we briefly describe our method for tool path generation [Sarma and Dutta, 1996a] using offset curves, for a single parametric patch. We assume ball ended tools and a 3-axis machine. The two steps involved are: (a) calculating the step-over and (b) sequentially generating the tool paths. The part is assumed to be a parametric patch S: $r(u,v)$ with unique normals $\hat{n}(u,v)$. Curves on the parametric patch are represented in terms of triples C: $\{u(t), v(t), h(t)\}$, where $u(t)$ and $v(t)$ represent a parametric curve on $r(u,v)$ and $h(t)$ is a distance function.

2.1 Offset Method for Tool Path Generation:

The first order approximation of step-over on the parametric patch S is calculated by taking offsets of a given curve C_i on S. The step-over calculations can be symbolically expressed by (EQ 1) and (EQ 2). If C_i: $\{u(t),v(t),h(t)\}$ is a scallop curve, then the tool path C_{i+1}: $\{u'(t),v'(t),0\}$ is obtained using (EQ 1). If C_i: $\{u(t),v(t),0\}$ is a tool path, then the scallop curve C_{i+1}: $\{u'(t),v'(t),h(t)\}$ is obtained using (EQ 2). The functions $U_t(t)$, $V_t(t)$, $U_s(t)$ and $V_s(t)$ represent a parametric direction of the offset. The functions $\lambda_t(t)$ and $\lambda_s(t)$ represent the parametric distance of the offset.

$$\begin{Bmatrix} u'(t) \\ v'(t) \end{Bmatrix} = \begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} + \lambda_s(t) \begin{Bmatrix} U_s(t) \\ V_s(t) \end{Bmatrix} \quad \text{(EQ 1)}$$

$$\begin{Bmatrix} u'(t) \\ v'(t) \end{Bmatrix} = \begin{Bmatrix} u(t) \\ v(t) \end{Bmatrix} + \lambda_t(t) \begin{Bmatrix} U_t(t) \\ V_t(t) \end{Bmatrix} \quad \text{(EQ 2)}$$

We refer to $h(t)$ as the scallop height function, and it is specified by the user. The selection of the type of scallop height function (i.e. constant, geodesic, normal and distance) is based on practical metrics such as length, curvature and number of tool paths. For more details please refer to [Sarma, 1996].

Tool path generation techniques, in general, attempt to compromise between conflicting objectives (reducing time and increasing accuracy). Our method consists of the following sequential steps based on (EQ 1) and (EQ 2): (i) given a tool path find the farthest scallop curve that satisfies a given scallop height function and (ii) given a scallop curve find the farthest tool path that lies on the parametric patch. The procedure begins by the user specifying a starting curve on the parametric patch.

Procedure:

1. Calculate the scallop height function $h(t)$ based on the local properties of the part S along the given starting curve C_0.
2. Using (EQ 1) or (EQ 2), compute C_1 from C_0. If C_0 is a scallop curve (tool path) then C_1 is a tool path (scallop curve).
3. If C_1 lies within the parametric bounds of S, then $C_0 \leftarrow C_1$ and proceed to Step 1. If C_1 lies outside the parametric bounds of S, then the tool paths span S and tool path generation is complete.

Several parts have been machined using this method and are reported in [Sarma, 1996].

2.2 Extensions to Multi-Patch Surfaces:

In general, geometrically complex parts are not designed with a single parametric patch. Hence there is a need to extend the single patch technique for tool path generation to handle parts made of multi-patch surfaces. In addition, we also consider integration with commercial CAD/CAM packages and describe methods for the same. In extending our method for tool path generation to cover parts with multi-patch surfaces, there are several issues that need to be addressed. They fall under the following categories:

1. inputs for tool path generation
2. curve data structure
3. calculation of adjacent curve.

Firstly, in going from single patch to multi-patch surfaces, the geometry and topology of the part needs to be known. This is addressed in Section 3. Next, the connectivity of the tool paths and scallop curves needs to be maintained for multi-patch surfaces, for which a data structure is proposed. The details of the data structure are described in Section 4. Finally, sub-problems related to calculation of the adjacent curve, relevant to multi-patch surfaces, need to be solved. These sub-problems are: (a) calculations of offsets, (b) calculating the offset across patch boundaries, (c) interpolation of parameter space curves, (d) extrapolation and trimming of curves when necessary, (e) accuracy of object space curves and (f) strategy to deal with faces that are not curvature continuous. We describe techniques for these in Section 5.

3.0 Inputs for Tool Path Generation:

The inputs required for our method of tool path generation are: (a) part, (b) tool radius, (c) maximum and minimum allowable scallop height, (d) initial part orientation (important for 3-axis machining), (e) type of scallop height function and (f) starting curve.

The part is specified as a solid or sheet body [UniGraphics, 1992] [Acis, 1995]. Bodies are the highest level entities in geometric modelers which are able to keep record of the topology and geometry of the part. The conceptual structures that we have been dealing with are as shown in Figure 2 where we are interested in the geometry of the faces and edges in order to generate tool paths. We make the assumption that all the faces in each body are joined with curvature continuity in the object space. We discuss exceptions to this assumption in Section 5.7.

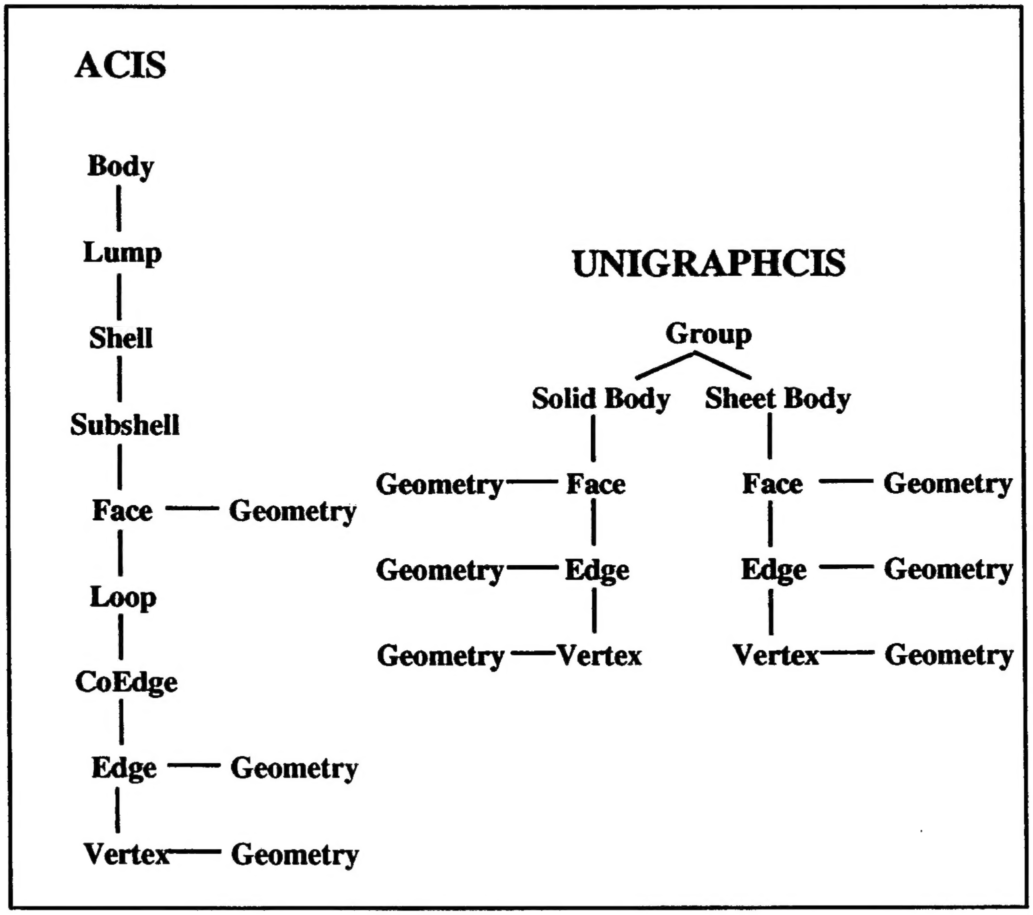

FIGURE 2. Illustration of ACIS and UNIGRAPHICS Data Structures of Interest

The tool, as mentioned earlier, is assumed to be ball ended. The highest curvature on the part corresponds to the smallest tool radius. A curvature analysis [Maekawa and Patrikalakis, 1994] of the part is performed to determine the tool radius. The maximum and minimum allowable values of the scallop height are specified by the user depending on the required accuracy of the manufactured part.

The initial part orientation is determined by examining the part normals. For 3-axis machining, it is required that all the normals of the machinable side of the part lie in one hemisphere. That part normal which makes an angle of no more than 90° with all the other normals is picked as the tool axis vector. This is illustrated in Figure 3.

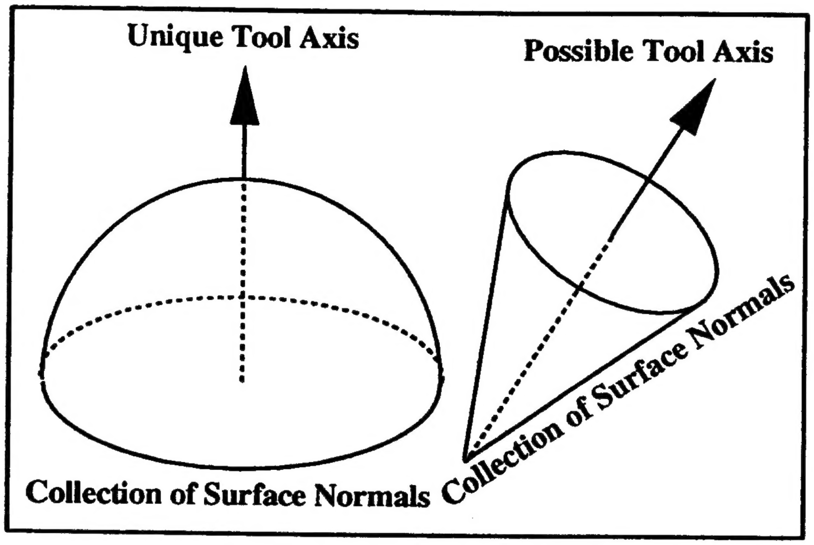

FIGURE 3. Selection of Tool Axis Given a Collection of Part Normals

The type of scallop height function (constant, geodesic, normal and distance) can be selected depending on whether length, curvature or number of the tool paths is a priority. The scallop height function is switched depending on the following criteria: (a) the constant scallop height function is the default scallop height function, (b) if the tool paths develop cusps and self-intersections (i.e. if (EQ 3), where κ_g and R are the geodesic curvature of the tool path and tool radius respectively, is close to being violated) the geodesic or normal scallop height functions can be used to compensate and (c) if the part geometry results in excessive tool path segmentation, the distance scallop height function can be used.

$$\kappa_g \le \frac{1}{R} \tag{EQ 3}$$

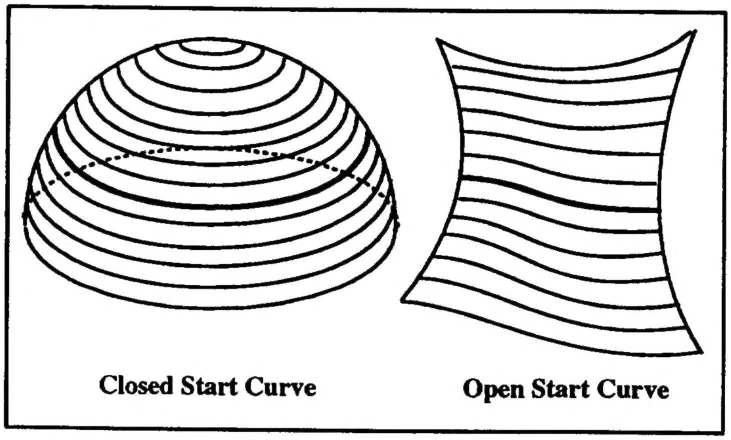

FIGURE 4. Selection of Start Curve for Different Parts

The starting curve can be specified by object space constraints such as (a) constraining the starting curve to lie on the intersection curve of a surface and the part and (b) constraining the starting curve to lie on the projection of a curve (or projection along a given direction) on the part. The starting curve can also be specified by parameter space constraints such as (a) specifying iso-parameter curves and (b) specifying curves in the parameter space via control points. In our experience, a boundary curve or a curve that is approximately in the center of the part is a good candidate for the starting curve. The starting curve could also be specified to be a closed or open curve depending on the shape of the part as shown in Figure 4.

4.0 Curve Data Structure:

An important component of our method for tool path generation for multiple patches, is maintaining the connectivity of the tool paths and scallop curves between adjacent patches. Figure 5 shows a part comprising of a grid of patches in the object space and in the parameter space. For example the curve shown in a dotted line in Figure 5 spans five surfaces. In order to traverse the curve, the "next" and "previous" segments of the curve have to be known. In addition, since we deal with parametric space curves as well as object space curves, the parent parametric patch has to be known.

This information is stored by means of a data structure that is schematically shown in Figure 6. The data structure keeps track of the previous curve (CURVE), next curve (CURVE), parametric space curve (PCURVE), object space curve (OCURVE) and the face (FACE). This data structure enables us to (a) maintain connectivity of tool paths and scallop curves, (b) keep information of parametric curves and (c) maintain correspondence between the object curves, parametric curves and faces.

5.0 Calculation of Adjacent Curve:

In this section we discuss details of (a) exact calculations of offsets, (b) calculating the offset across patch boundaries, (c) interpolation of parameter space curves, (d) extrapolation and trimming of curves when necessary, (e) accuracy of object space curves and (f) strategy to deal with faces that are not curvature continuous.

Of relevance in the offset calculation, is the shape of the tool in the vicinity of the part. First, the cutting tool is replaced by a sphere of the same radius. This simplifies the computation of the swept section of the tool necessary in our tool path generation. The swept sections (i.e. cross sections of the swept volume) are circles that lie in the normal plane (i.e. plane spanned by the tangent and normal of the tool path at any given point) of the part.

5.1 Calculation of Tool Path From Scallop Curve:

Consider a tool path C_i: $\{u(t), v(t), 0\}$ on a single parametric patch S: $r(u,v)$. The goal is to calculate the scallop curve C_{i+1}: $\{u'(t), v'(t), h(t)\}$, where the scallop height function $h(t)$

is known. Let us consider a point P_t on C_i evaluated at the parameter t_0. Let a swept section Π be placed at P_t. Let the corresponding point on the scallop curve be P_s. The points P'_t and P'_s are points at offsets of the tool radius (R) and scallop height (h_0) respectively from the parametric patch S. This is illustrated in Figure 7. Note that while Figure 7 is an illustration in two dimensions, the point P_s does not necessarily lie in the plane of Π. By definition of the swept section, the unit normal $\hat{t}_0$ of the plane Π can be calculated as follows:

$$\hat{t}_0 = \frac{r_{u0}u_{t0} + r_{v0}v_{t0}}{|r_{u0}u_{t0} + r_{v0}v_{t0}|} \qquad \text{(EQ 4)}$$

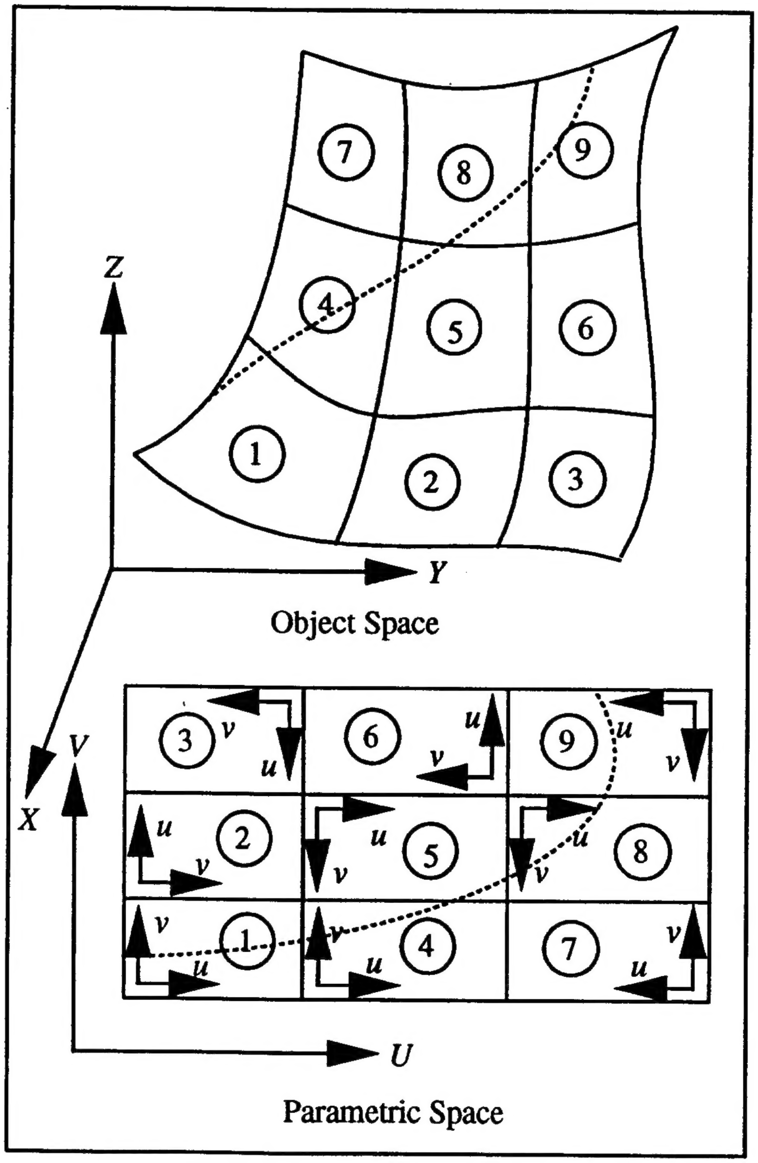

FIGURE 5. Illustration of a Curve in the Parametric and Object Spaces

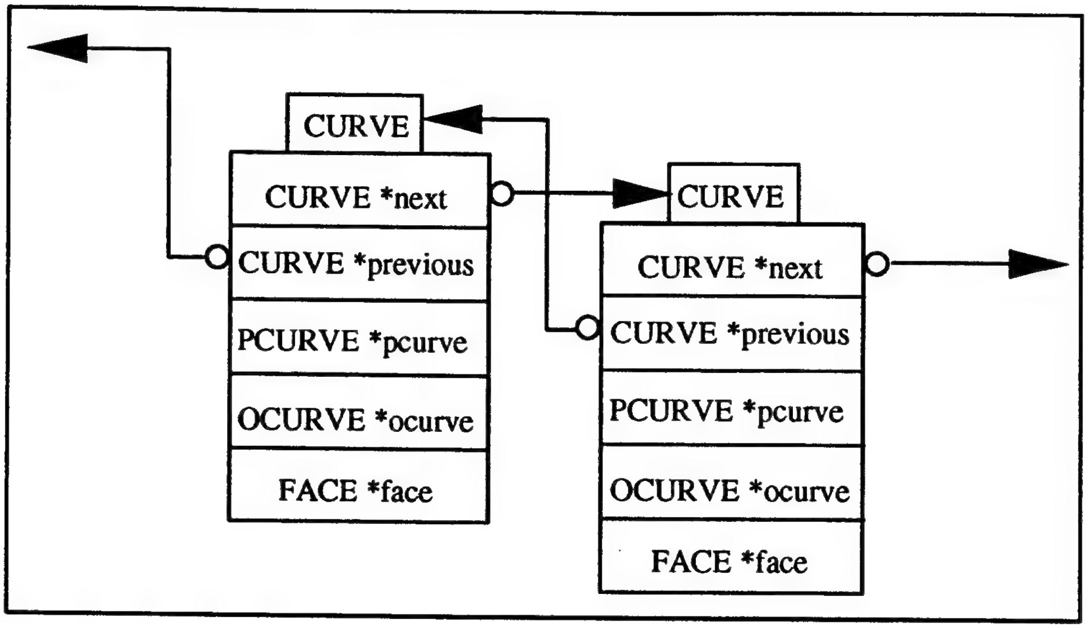

FIGURE 6. Data Structure for Scallop Curve and Tool Path

where it is assumed that the subscript *0* implies evaluation at the parameter t_0 and the other subscripts imply partial differentiation with respect to the variable. The plane of the swept section is spanned by the unit vectors $\hat{\boldsymbol{n}}_0$ and $\hat{\boldsymbol{b}}_0$, where $\hat{\boldsymbol{n}}_0$ is the unit normal of the parametric patch evaluated at t_0 and $\hat{\boldsymbol{b}}_0$ is a vector orthogonal to both $\hat{\boldsymbol{n}}_0$ and $\hat{\boldsymbol{t}}_0$. The point $\boldsymbol{P_s}$ can be found by evaluating the point at which the residual $\Re$ shown below vanishes.

$$\Re = h - \left|\boldsymbol{r}_0 + R\hat{\boldsymbol{n}}_0 + RCos\theta\hat{\boldsymbol{b}}_0 - R\sin\theta\hat{\boldsymbol{n}}_0 - Proj(\boldsymbol{r}_0 + R\hat{\boldsymbol{n}}_0 + RCos\theta\hat{\boldsymbol{b}}_0 - R\sin\theta\hat{\boldsymbol{n}}_0)\right| \qquad \text{(EQ 5)}$$

The operator ***Proj(P)*** represents normal projection of the point $\boldsymbol{P}$ on the surface $\boldsymbol{S}$. (EQ 5) can be solved by using any iterative technique e.g. modified regula falsi, etc., to obtain the point $\boldsymbol{P_s}$ and its parameter values on the surface $\boldsymbol{S}$ (the parameter values are a by-product of the normal projection). The starting point for the iterative calculation can be specified as the first order approximation shown in Section 2.1. It is possible that $\boldsymbol{P_t}$ and $\boldsymbol{P_s}$ do not lie on the same parametric patch. Then, the normal projection of $\boldsymbol{P}$ is considered with respect to the parametric patch of $\boldsymbol{P_s}$. The problem of detecting the jump in parametric patches is addressed in Section 5.3.

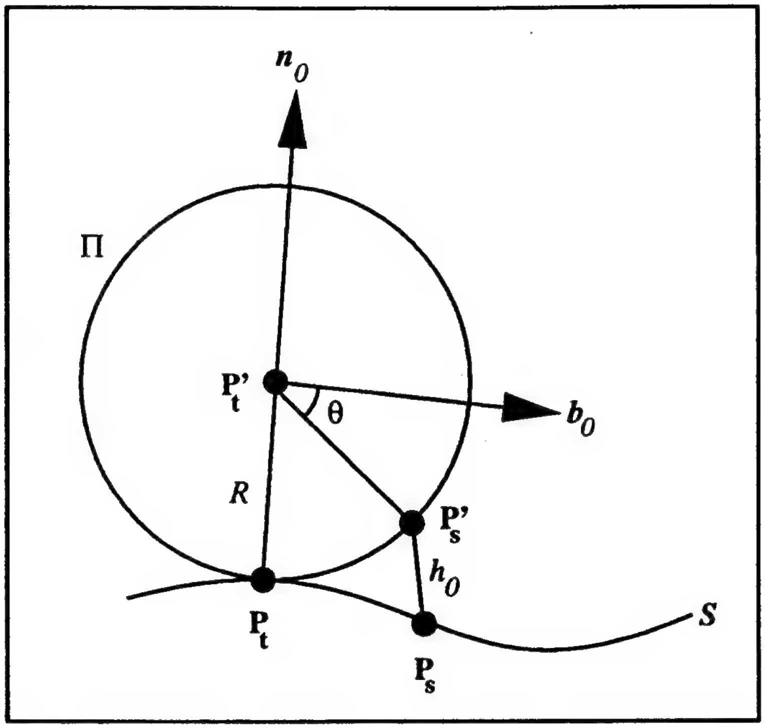

FIGURE 7. Simplified Illustration of Swept Section Located on a Parametric Patch

5.2 Calculation of Scallop Curve From Tool Path:

Consider a scallop curve C_i: $\{u(t), v(t), h(t)\}$ on a single parametric patch S: $r(u,v)$, where the function $h(t)$ is known. The goal is to calculate the tool path C_{i+1}: $\{u'(t), v'(t), 0\}$. Let us consider a point P_s on C_i evaluated at the parameter t_0. Let the corresponding point on the tool path be P_t. The points P'_t and P'_s are points at offsets of the tool radius (R) and scallop height (h_0) respectively from the parametric patch S. Consider a circle Π which is placed at P'_s such that its plane normal is the tangent of the scallop curve. This is illustrated in Figure 8 where, by definition of the swept section, P'_t is constrained to lie on Π (the dashed circle). Note that Figure 8 is a two dimensional illustration and P_t does not necessarily lie in the plane of Π.

$$\hat{t}_0 = \frac{r_{u0}u_{t0} + r_{v0}v_{t0} + h_0\hat{n}_{u0}u_{t0} + h_0\hat{n}_{v0}v_{t0}}{\left|r_{u0}u_{t0} + r_{v0}v_{t0} + h_0\hat{n}_{u0}u_{t0} + h_0\hat{n}_{v0}v_{t0}\right|} \qquad \text{(EQ 6)}$$

The plane of the circle Π is spanned by the unit vectors $\hat{n}_0$ and $\hat{b}_0$, where $\hat{n}_0$ is the unit normal of the parametric patch evaluated at t_0 and $\hat{b}_0$ is a vector orthogonal to both $\hat{n}_0$ and $\hat{t}_0$. The point P_t can be found by evaluating the point at which the residual $\Re$ shown below vanishes.

$$\Re = R - \left| r_0 + h_0\hat{n}_0 + RCos\theta\hat{b}_0 + R\sin\theta\hat{n}_0 - Proj(r_0 + h_0\hat{n}_0 + RCos\theta\hat{b}_0 + R\sin\theta\hat{n}_0) \right| \quad \text{(EQ 7)}$$

Once again, (EQ 7) can be solved by an iterative numerical method for which the starting point can be specified by the first order approximation given in Section 2.1. It is possible that P_t and P_s do not lie on the same parametric patch. Then, the normal projection of P is considered with respect to the parametric patch of P_t.

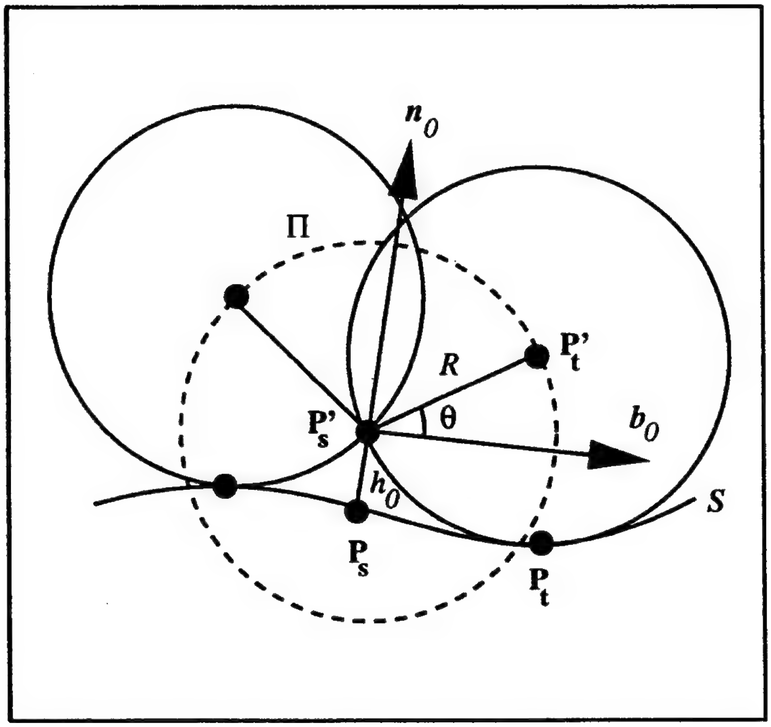

FIGURE 8. Simplified Illustration of Swept Section Located on a Parametric Patch

5.3 Offsetting at Patch Boundaries:

Figure 9 shows a situation where it will be required to calculate the offset across parametric patch boundaries. For example, curve A spans the parametric patches 1, 4, 8, 9 and 6, while its offset B spans 1, 4, 7, 8, 9 and 6. The change in parametric patches is detected when the first order approximation for calculating offsets (Section 2.1) yields an offset point outside the current parametric patch. This happens when the parametric coordinates of the offset point lie outside the parametric domain of the current patch. There are three cases in which the offset point lies outside the current surface as illustrated in Figure 10.

Case (a) can be detected when the curve A is locally very close to the boundary of the current surface *S*. The closeness is measured when parametric distances from consecutive points on A to the boundary of *S* are less than a pre-specified parametric distance tolerance. This usually occurs when the start curve approaches the end of the parametric domain of the current surface *S*. In this case, the edge nearest to the start curve A, such as

E is detected and the surface *S*' adjacent to E and distinct from *S* is found. This surface *S*' is then used as a candidate for projecting points on the offset curve.

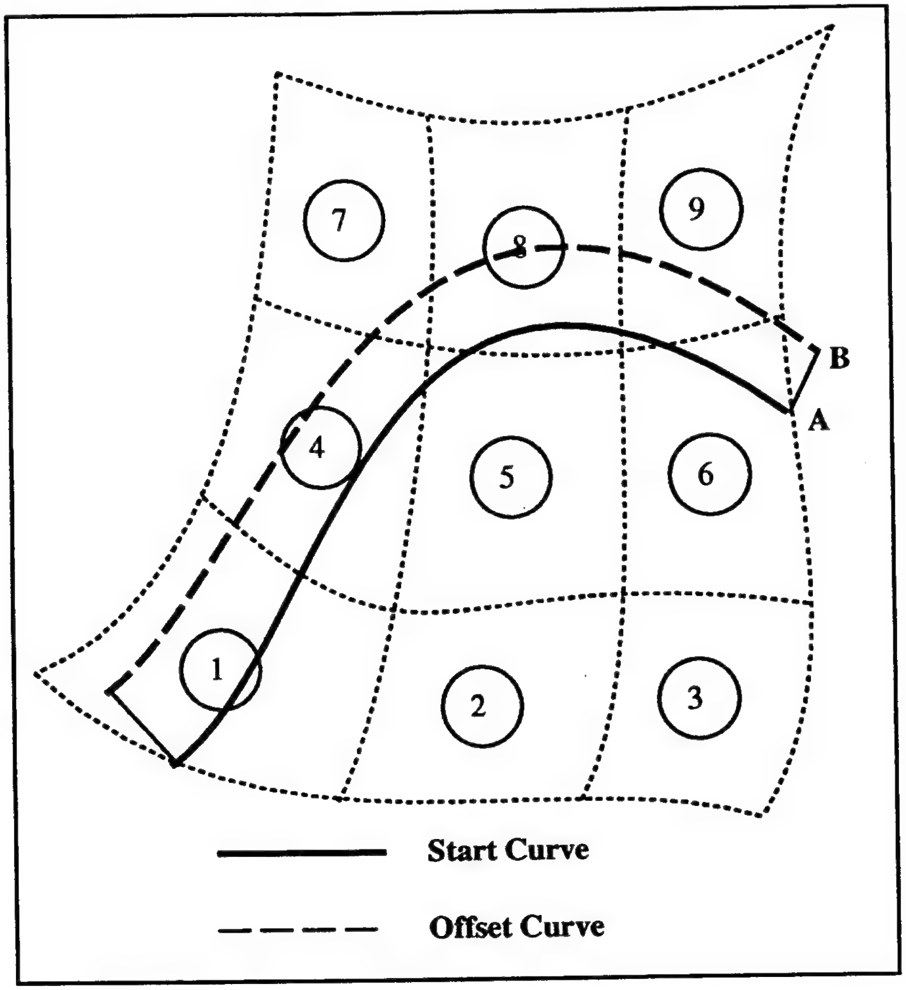

FIGURE 9. Illustration of Offsets Across Patch Boundaries

Case (b) can be detected when the end points of curve A on consecutive surfaces *S* and *S*' are coincident in the object domain. This is the easiest situation to detect, since the start and end points of the start curve A on each surface *S* and *S*' can be calculated (from the curve data structure). In addition, only one edge such as E can be found in the vicinity of the coincident points where there is a change in surfaces from *S* to *S*'. In such a case, the surface *S*' adjacent to E and distinct from *S*, is used as a candidate for projecting the points on the offset curve.

Case (c) can be detected when curve A locally spans three surfaces *S*, *S*' and *S*''. In addition, the length of one segment of curve A (e.g. the segment of curve A on *S*') is lesser than a pre-specified parametric length tolerance. Let *S*' be the current surface. In such a case, the vertex V closest to the segment of curve A on *S*' is found. In addition, the edges E1 and E2 adjoining the vertex V on the current surface *S*' are also found. The candidate surface for projection is that surface which shares the vertex V with *S*', but does not share edges E1 and E2 with *S*'. If more than four surfaces are connected at vertex V, then all the surfaces that satisfy the above condition are candidates for projecting points.

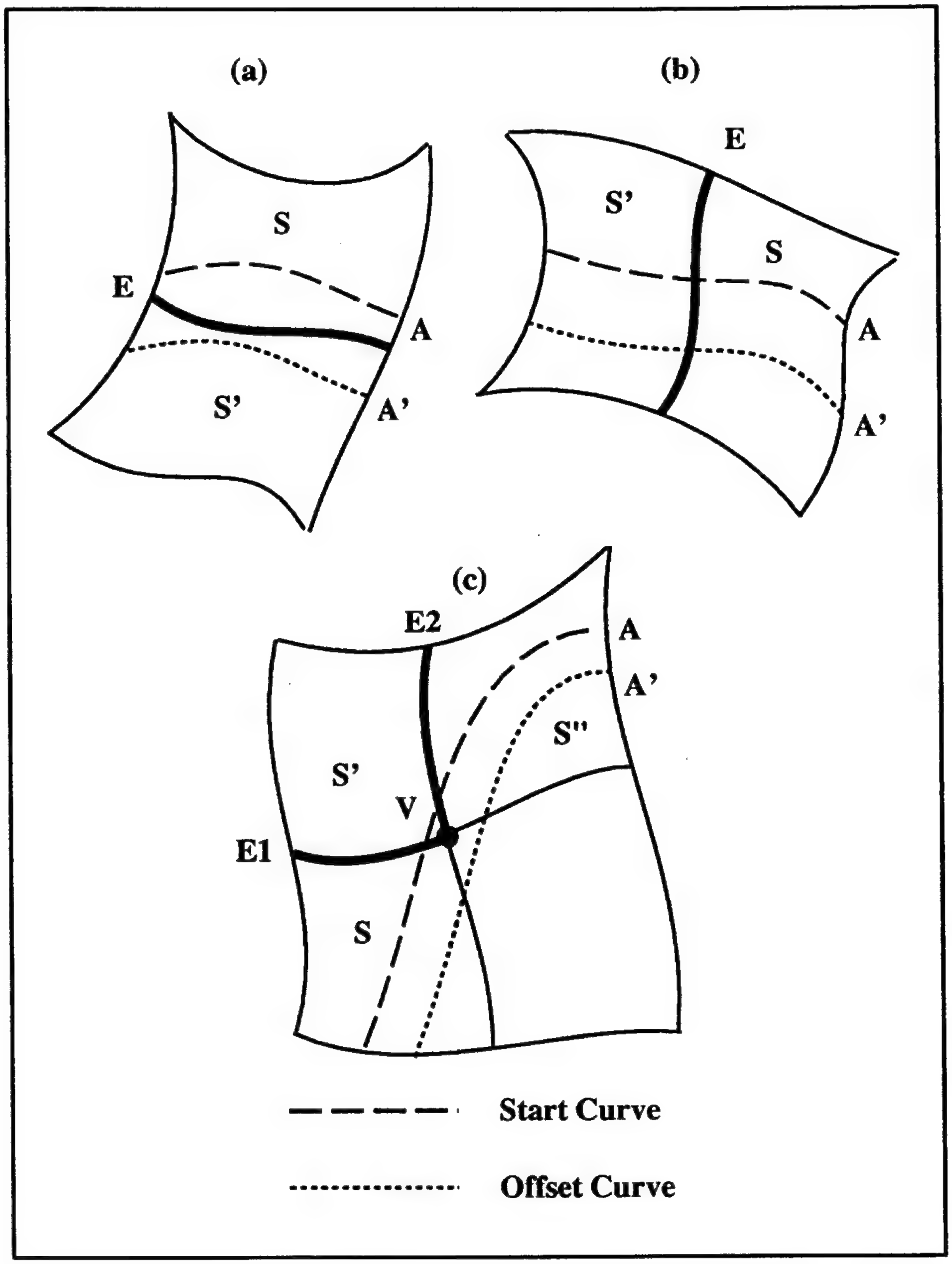

FIGURE 10. Illustration of Offsets at Patch Boundaries

5.4 Interpolation of Parameter Space Curves:

The result of offsetting a given curve on the parametric patches is a set of parameter space points with their corresponding parametric patches, which have to be interpolated to obtain another set of parametric curves. Since we are interested in the offset property of the corresponding object space curves, we use a Hermite interpolation of the parameter space points to obtain the parameter space curves. Hermite interpolation requires the use of points and tangents. Please refer to any standard textbook [Farin, 1990] for details of

Hermite interpolation. In this section we outline a method to determine the tangents at the points of interest.

First we describe the calculation of tangents of the scallop curve given the tangents of the tool path. We refer back to Figure 7 which shows the swept section Π with respect to the part. Consider the tangent to the scallop curve represented by the vector $\hat{T}_o$ which will have an equation as shown in (EQ 6). By definition of the swept section, the tangents at the points P_t and P'_s are required to be unit vectors of the same direction given by (EQ 6). Hence we have the following equation:

$$\hat{T}_0 \cdot \hat{b}_0 = 0 = \left(\frac{r_{u0}\delta u + r_{v0}\delta v + h_0\hat{n}_{u0}\delta u + h_0\hat{n}_{v0}\delta v}{\left|r_{u0}\delta u + r_{v0}\delta v + h_0\hat{n}_{u0}\delta u + h_0\hat{n}_{v0}\delta v\right|}\right) \cdot \hat{b}_0 \qquad \text{(EQ 8)}$$

This equation can be solved to yield the unit parameter space tangents as follows, where the parametric space tangents have to be normalized.

$$\begin{Bmatrix} \delta u \\ \delta v \end{Bmatrix} = normalize \begin{Bmatrix} -\hat{b}_0 \cdot (r_{v0} + h_0\hat{n}_{v0}) \\ \hat{b}_0 \cdot (r_{u0} + h_0\hat{n}_{u0}) \end{Bmatrix} \qquad \text{(EQ 9)}$$

Now we describe the calculation of tangents of the tool path given the tangents of thescallop curve. We refer back to Figure 8 which shows the circle Π with respect to the part. Consider the tangent to the scallop curve represented by the vector $\hat{T}_0$ which will have an equation as shown in (EQ 4). By definition of the circle Π, the tangents at the points P'_s and P_t are required to be unit vectors of the same direction given by $\hat{t}_0$. Hence we have the following equation:

$$\hat{T}_0 \cdot \hat{b}_0 = 0 = \left(\frac{r_{u0}\delta u + r_{v0}\delta v}{\left|r_{u0}\delta u + r_{v0}\delta v\right|}\right) \cdot \hat{b}_0 \qquad \text{(EQ 10)}$$

This equation can be solved to yield the unit parameter space tangents as follows:

$$\begin{Bmatrix} \delta u \\ \delta v \end{Bmatrix} = normailze \begin{Bmatrix} -\hat{b}_0 \cdot r_{v0} \\ \hat{b}_0 \cdot r_{u0} \end{Bmatrix} \qquad \text{(EQ 11)}$$

5.5 Extrapolation and Trimming:

Often the tool paths (scallop curves) need to be trimmed or extrapolated to span the entire part. This is shown in Figure 9 where curve B' has to be extrapolated at one end and trimmed at the other end.

The extrapolation is done by extending the tangent a finite amount in the direction of the last tangent. This is shown in Figure 11 where an extra point **p** is introduced to the existing set of points and tangents. The tangent at the point **p** is the same as its direction of extension i.e. the extrapolation is linear.

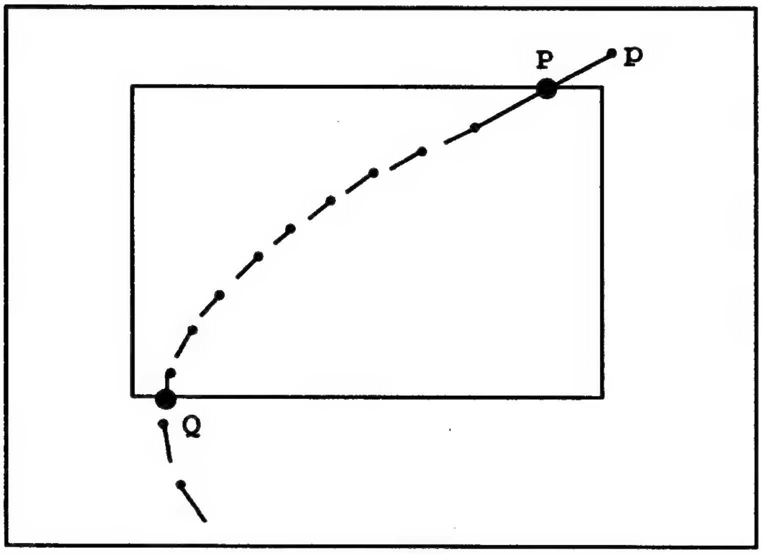

FIGURE 11. Illustration of Extrapolation and Trimming in Parameter Space

After all extrapolations are performed, the parameter space curve is obtained by Hermite interpolation. The resultant curve is then trimmed by intersecting it with each boundary of the parametric patch. Note that this operation is a curve-curve intersection in a plane. For example, in Figure 11, the points **P** and **Q** are determined by intersecting the curve interpolated through the points, with the parametric curve boundaries.

5.6 Accuracy of Object Space Curves:

Deriving the object space curves from the parameter space curves is necessary since the tool paths and scallop curves are actually in object space. In addition, any subdivision of the parameter space curves has to be done by considering the discretization errors in the object space. We evaluate points on the object space curve such that the object space discretization error is lesser than a maximum allowable value ε. This requires (a) the calculation of the curvature of the object space curve and (b) evaluating points on the object space curve based on the discretization error.

The curvature of a curve on a surface consists of two components - the normal curvature and the geodesic curvature. The normal curvature is that component of the curvature contributed by the surface on which the curve lies. The geodesic curvature is that component of curvature contributed by the curve meandering in the tangent space of the surface and hence is sometimes also referred to as the tangential curvature.

Consider a parameter space curve *C* represented by the functions $\{u(t), v(t), 0\}$. Let the object space tangent of the curve *C* be represented by the vector $\hat{t}$. Assume that the corresponding object space curve is parametrized by its arc-length s. The normal curvature vector κ_n and the geodesic curvature vector κ_g are given by the following expressions [Struik, 1950]:

$$\kappa_n = \hat{t}_s \qquad \text{(EQ 12)}$$

$$\kappa_g = (\hat{n} \times \hat{t}) \cdot \hat{t}_s \qquad \text{(EQ 13)}$$

$$\kappa = \kappa_n + \kappa_g \qquad \text{(EQ 14)}$$

where $\hat{n}$ is the unit normal vector of the surface and $\hat{t}_s$, using the same convention as in Section 5.1, is the derivative of the unit tangent vector with respect to the arc length s. The curvature vector of the curve C in object space is a vector sum of the normal and geodesic curvature vectors as shown in (EQ 14). The expression for $\hat{t}_s$, in a form easy for evaluation can be obtained by applying chain rules of differentiation as follows:

$$\hat{t}_s = t_s^2 r_{tt} + \frac{r_t \cdot r_{tt}}{t_s^{\frac{1}{2}}} r_t \qquad \text{(EQ 15)}$$

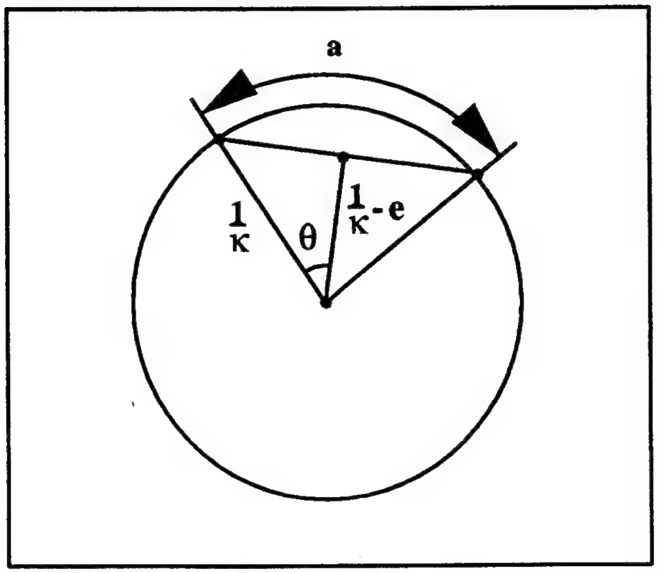

FIGURE 12. Illustration of Discretization Based on Circle of Curvature

where t_s is the derivative of the parameter t with respect to the arc length given by:

$$t_s = \frac{1}{\left|r_u u_t + r_v v_t\right|} \qquad \text{(EQ 16)}$$

The curve C can be discretized by considering the circle of curvature of the object space curve as shown in Figure 12 and stepping along C by the arc-length a given by the following expression:

$$a = \frac{2}{\kappa}\theta = \frac{2}{\kappa}\operatorname{acos}(1-\kappa\varepsilon) \qquad \text{(EQ 17)}$$

5.7 C^0 and C^1 Continuous Patches:

In the case of parametric patches which do not have curvature continuity in the object space, the tool paths and scallop curves develop discontinuities as shown in Figure 13. This is disadvantageous for two reasons: (a) gouging or over-machining could occur and (b) the length of the tool paths could increase depending on the connectivity of tool paths. We have developed heuristic procedures to prevent gouging and inefficient tool path connectivity.

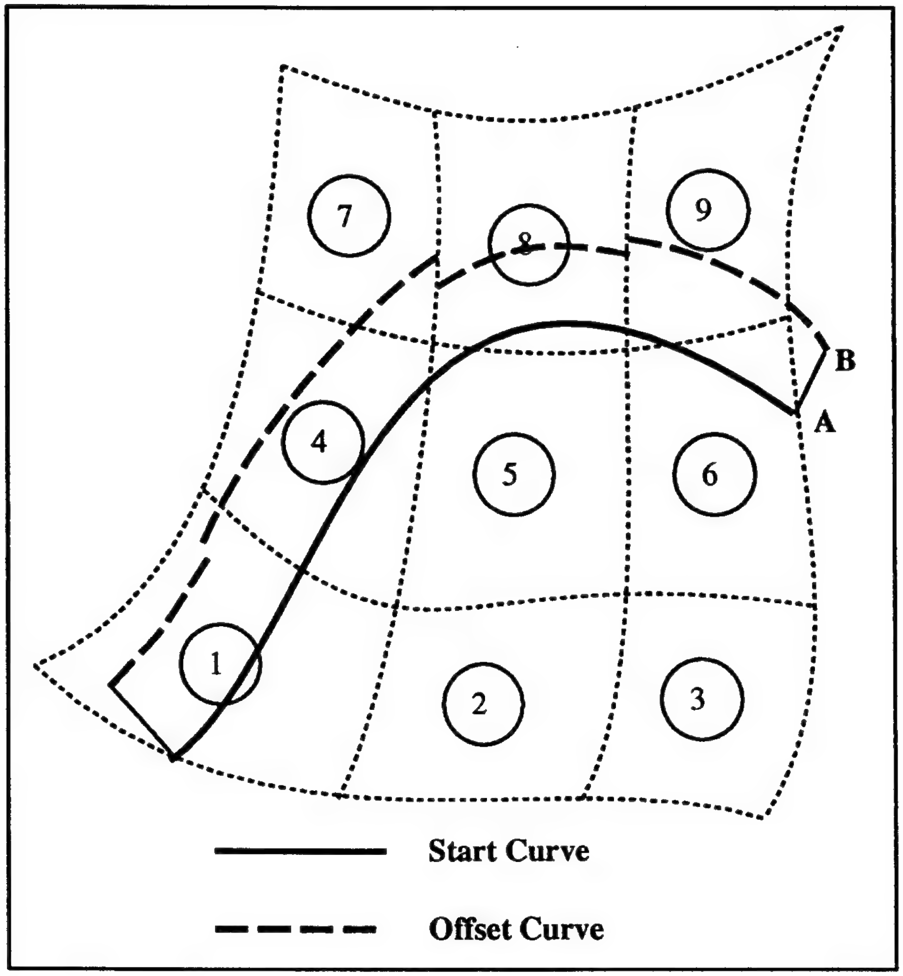

FIGURE 13. Illustration of Offsets Across Curvature Discontinuous Patches

First the part is divided into regions which are curvature continuous. If the curvature continuous regions are large enough compared to the part, tool path generation can be done independently in each curvature continuous region. If the curvature continuous regions are small compared to the part, the following heuristics are applied.

Case (a): If adjacent surfaces are curvature continuous, yet yield discontinuous tool paths/ scallop curves as shown in Figure 14a (left), we conclude that there are numerical errors in calculations. In such cases we join the segments of the offset curve as shown in Figure 14a (right). This joining is performed by (i) extrapolating each curve segment in the surface domain such that it locally follows the adjacent curve segment and (iii) trimming the parametric curve segments.

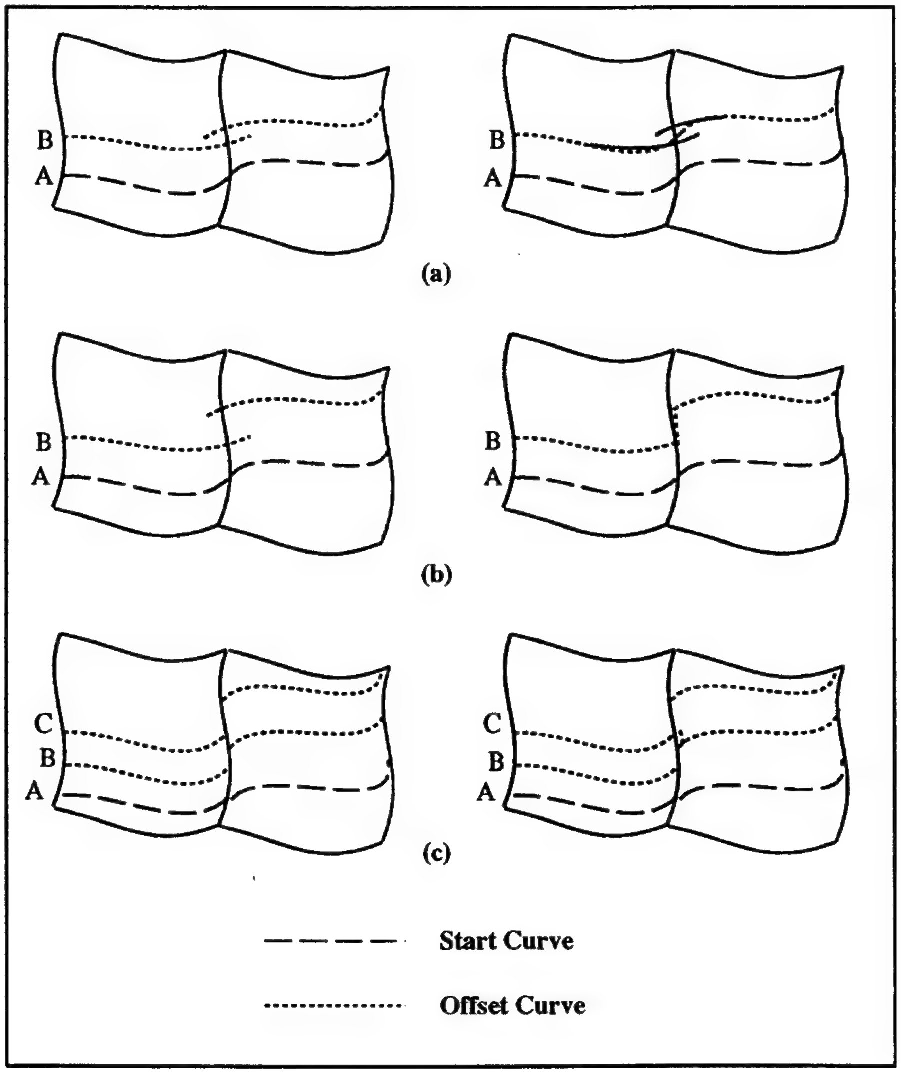

FIGURE 14. Heuristics for Curvature Discontinuous Surfaces

Case (b): If adjacent surfaces are curvature discontinuous as illustrated in Figure 4.30b (left) the offset curves of A on each surface cannot be joined by interpolation alone. We propose the following steps: (a) trim the offset curves and (b) introduce an intermediate curve segment, along the common boundary between adjacent surfaces, joining the discontinuous segments of the tool path/scallop curve as shown in Figure 4.30b (right). The intermediate curve segments need to be tagged, since they need not be offset any further - only the trimmed curves segments of B are candidates for further offsetting.

Case (c): If adjacent surfaces are curvature discontinuous as illustrated in Figure 4.30c (left), and joining the corresponding offset curve segments by intermediate curves causes significant over-machining on the common boundary between the surfaces, the following steps are proposed: (a) trim the offset curves and (b) introduce an intermediate curve segment, along the common boundary between adjacent surfaces, joining the discontinuous segments of the tool path/scallop curve as shown in Figure 4.30c (right). In this case the intermediate curve segment need not join segments of the same offset curve. For example, in Figure 4.30c (right) an intermediate curve is introduced between a segment of curve C and another segment of curve B.

6.0 Integration with Native System and Examples:

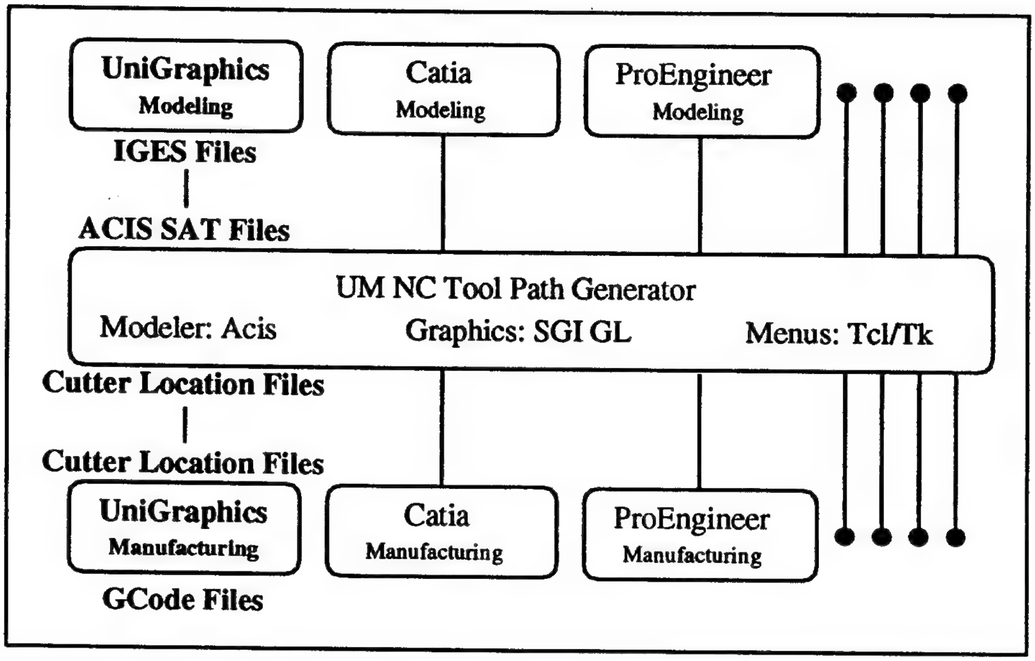

FIGURE 15. Illustration of Integration with Native CAD/CAM System

Our method for generating tool paths has been implemented on the Silicon Graphics Personal Iris workstation in C++. The ACIS geometric modeler has been used for generating tool paths. The user interface for our implementation uses the Tcl/Tk and GL libraries. Figure 15 illustrates a system architecture for integration with native CAD/CAM systems such as UniGraphics, Catia and ProEngineer. In our case, the link between the CAD and CAM modules has been established for UniGraphics by (a) using a module built on the Acis geometric modeler and (b) using the native systems' programming languages.

Figure 16 shows an example of an intake manifold made of a grid of 3x2 parametric patches. The inatke manifold was designed in the UniGraphics modeler. A maximum scallop height (h_{max}) of 0.004 inches and a minum scallop height (h_{min}) of 0.002 inches were specified. The tool paths were generated using UniGraphics' GRIP programming language. The length of the tool paths (*l inches*), the maximum geodesic curvature of the tool paths (κ *inches* $^{-1}$), and the number of tool paths (*N*) for different types of tool paths are listed in Table 1 for the same tool radius (*R inches*). The CL (cutter location) file generated was post-processed using the UniGraphics manufacturing module.

Type of Tool Path	l	κ	N	$h_{\max}$	$h_{\min}$	R
Iso-Parametric	195.3	0.031	34	0.004	-	0.25
Constant Scallop Height	139.9	0.047	33	0.004	0.004	0.25
Geodesic Scallop Height	165.4	0.026	36	0.004	0.002	0.25
Distance Scallop Height	150.4	0.030	33	0.004	0.002	0.25

TABLE 1. Statistics For Different Tool Paths for the Intake Manifold

The tool paths generated above were judged with respect to the criteria of length, maximum geodesic curvature and number of tool paths. These criteria were selected, as explained in detail in [Sarma and Dutta, 1996a], for the following reasons: (a) the time of machining is proportional to the length of the tool paths, (b) the size of the CL file and the time of manufacturing depends on the maximum curvature of the tool paths and (c) the number of sharp turns and/or tool retractions in the tool path is proportional to the number of tool paths. The constant scallop height tool paths were judged to be most suitable for the intake manifold due to a good compromise in the length, curvature and number of tool paths. A mold was manufactured using the constant scallop height tool paths on a FADAL 3-axis NC milling machine. The final part was cast using an epoxy and is shown in Figure 17.

7.0 Summary:

In this paper, we have described a system for tool path generation for multi-patch surfaces, since parts in the industry are usually designed with multi-patch surfaces. We described several issues in tool path generation that arise when considering parts with multiple patches/faces such as (a) offsetting across patches, (b) extrapolation, interpolation and trimming of parameter space curves, (c) accuracy of object space curves and (d) curvature discontinuous patches. Each of these were implemented using (i) the Acis geometric modeler and (ii) UniGraphics' programming language GRIP. We also describe a method for integration of the proposed method for tool path generation with commercial CAD/CAM systems. Our ongoing work involves extending our techniques to patches with arbitrary trimming curves. Our future work involves extension of our method for tool path generation for flat ended tools.

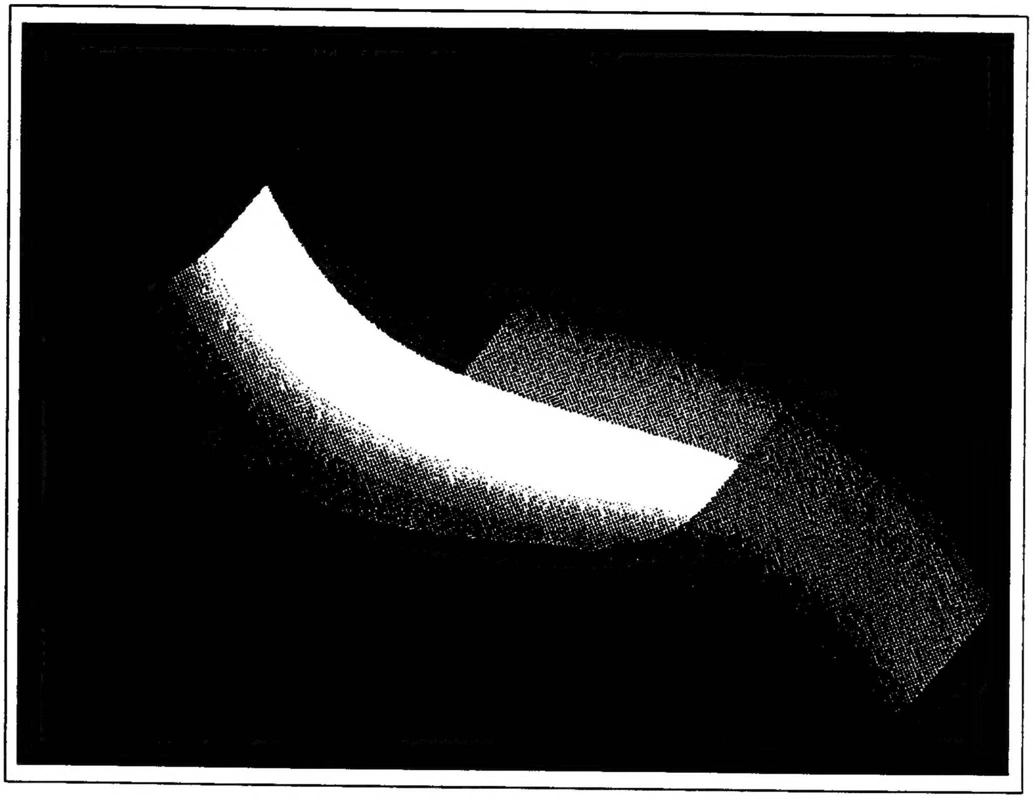

FIGURE 16. Example of Tool Paths Generated for an Intake Manifold With Multiple Patches

FIGURE 17. Intake Manifold Manufactured Using Constant Scallop Heights

8.0 Acknowledgments:

The financial support from AFOSR grant number F49620-95-1-0209 is gratefully acknowledged.

9.0 Bibliography:

Acis, Spatial Technology Inc. Online Documentation Version 2.0, 1995.

Bobrow, J. E., Solid Modelers Improve NC Machine Tool Path Generation Techniques, Computers in Engineering, Vol. 1, 1985, pp. 439-444.

Beck, J. M., Farouki, R. T. and Hinds, J. K., Surface Analysis Methods, IEEE Computer Graphics and Applications, Vol. 6, N0. 12, December 1986, pp. 18-36.

Catia, Catia Manufacturing Solutions User Manual, Version 4, 1994.

Farin, G., Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press Inc., 1990.

Huang, Y. and Oliver, J. H., Non-Constant Parameter NC Tool path Generation on Sculptured Surfaces, International Journal of Advanced Manufacturing Technology, Vol. 9, 1994, pp. 281-290.

Jensen, C. G. and Anderson, D. C., Accurate Tool Placement and Orientation for Finish Surface Machining, Concurrent Engineering, PED-Vol 59, ASME 1992, pp. 127-145.

Koepfer, C., Machine Tool Considerations Come to the Surface, Modern Machine Shop, October 1993, pp. 62-73.

Lai, J-Y. and Wang, D-J., Strategy for finish cutting Path Generation of Compound Surfaces, Computers in Industry, Vol. 25, No. 2, December 1994, pp. 189-209.

Loney, G. C. and Ozsoy, T. M., NC Machining of Free Form Surfaces, Computer Aided Design, Vol. 19, No. 2, March 1987, pp. 85-89.

Maekawa, T. and Patrikalakis, N. M., Interrogation of Differential Geometry Properties for Design and Manufacture, Visual Computer, Vol. 10, No. 4, 1994, pp. 216-237.

ProEngineer, Pro Manufacture User's Guide, Version 15.0, 1995.

Sarma, R. and Dutta, D., Geometry and Generation of NC Tool Paths, To appear in the Journal of Mechanical Design, 1996a.

Sarma, R. and Dutta, D., Machining Surfaces Composed of Cyclide Patches, To appear in Computers in Industry, 1996b.

Sarma, R., NC Tool Path Synthesis, PhD. Dissertation, 1996, The University of Michigan, Ann Arbor, Michigan.

Struik, D. J., Lectures on Classical Differential Geometry, Dover Publications Inc., 1951.

Suresh, K. and Yang, D. C. H., Constant Scallop Height Machining of Free Form Surfaces, Journal of Engineering for Industry, Vol. 116, May 1994, pp. 253-259.

UniGraphics, EDS UniGraphics Online Documentation Version 10.4, 1992.